

AMENDMENT TO THE CLAIMS

1. (Previously Presented) A command processor stored on a computer readable memory for use with a computer system comprising:

- a graphical user interface program for providing a graphical interface to the computer system; and
- a command interpreter, which loads one or more configuration commands into the command processor from at least one of a user specified command configuration script comprising the one or more configuration commands or from a command line interface in which the one or more configuration commands are entered by a user, interprets the configuration commands and modifies the graphical user interface at run time of the graphical user interface according to the interpreted configuration commands, including:
 - building graphical objects within the graphical interface according to the interpreted configuration commands;
 - assigning functionality to the built graphical objects according to the interpreted configuration commands; and
 - displaying the graphical objects within the graphical interface according to the interpreted configuration commands;wherein the graphical objects are selectable by a user to access the assigned functionality to produce an integrated circuit design.

2-4. (Canceled)

5. (Original) The command processor of claim 1 and further comprising:
a suite of integrated circuit design tools, each design tool of the suite having a functionality corresponding to one or more steps in a design flow process of an integrated circuit.

6. (Previously Presented) The command processor of claim 5 wherein the command processor loads each design tool into the graphical user interface based on the user configuration commands.

7. (Previously Presented) The command processor of claim 1 and further comprising: a graphics engine tool for drawing contents of a database into the graphical user interface based on the user configuration commands.

8. (Currently Amended) A method of providing a fully customizable graphical user interface comprising:

upon execution of a command processor by a user, loading a top level Tool Command Language (TCL) command into a namespace, the command processor including a graphical user interface (GUI) without graphical objects;

loading one or more TCL commands into the command processor ~~from at least one of a user specified TCL command configuration script comprising the one or more TCL configuration commands or from a command line in which the one or more TCL configuration commands are entered by the user;~~

building graphical objects according to the TCL configuration commands;

assigning functionality to the built graphical objects according to the TCL configuration commands; and

displaying the graphical objects within the GUI according to the TCL

configuration commands, the graphical objects selectable by [[a]] the user to produce an integrated circuit design.

9. (Original) The method of claim 8 and further comprising:
performing functions based on user interactions with the graphical objects according to their
assigned functionality.
10. (Original) The method of claim 8 wherein the graphical objects are selected from a
group consisting of windows, window panes, buttons, and menus.
11. (Previously Presented) The method of claim 8 wherein the step of assigning
comprises:
creating the TCL command configuration script corresponding to a circuit design
function; and
assigning the TCL command configuration script to one of the graphical objects.
12. (Original) The method of claim 11 wherein the one of the graphical objects is a
button.
13. (Original) The method of claim 11 wherein the one of the graphical objects is an item
within a pull-down menu.
14. (Original) The method of claim 8 and further comprising:
changing a look and feel of the graphical user interface during a circuit design process.
15. (Previously Presented) The method of claim 14 wherein the step of changing
comprises:
creating new graphical objects, previously undefined by the command processor, using the
TCL configuration commands; and
assigning functionality to the new graphical objects.

16. (Previously Presented) The method of claim 14 wherein the step of changing comprises:

loading a new top level TCL command into the namespace, which corresponds to one or more new TCL configuration commands;

building graphical objects according to the new TCL configuration commands;

assigning functionality to the built graphical objects according to the new TCL configuration commands; and

displaying the user-interactive window containing the graphical objects according to the new TCL configuration commands.

17. (Canceled)

18. (Canceled)

19. (Previously Presented) A method of providing a graphical user interface, the method comprising:

loading a top level Tool Command Language (TCL) command into a namespace upon execution of a command processor;

providing a command interpreter for interpreting one or more configuration commands from a user;

loading a configuration command of the one or more configuration commands into the command processor from at least one of a user specified command configuration script comprising the configuration command or from a command line in which the configuration command is entered by the user; and

assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, the at least one graphical user interface (GUI) component selectable by a user to access an associated function to generate an integrated circuit design;

wherein all objects within the graphical user interface are user defined through the one or more configuration commands.

20. (Previously Presented) The method of claim 19 and further comprising:

changing the graphical user interface based on changing configuration commands from the user; and

displaying a changed graphical user interface during operation based on the changing configuration commands.

21. (Original) The method of claim 19 and further comprising:

interfacing with a suite of integrated circuit design tools for producing a integrated circuit layout and associated netlist.

22. (Original) The method of claim 21 wherein the step of interfacing comprises:
loading a design tool from the suite of design tools into the graphical user interface based on
a user command.

23. (Previously Presented) The method of claim 22 wherein the one or more user
configuration commands are assigned to one or more graphical objects.

24. (Previously Presented) An integrated circuit software design suite stored on a
computer readable memory and comprising:

a command processor having a graphical user interface and a command interpreter for
interpreting user commands, the graphical user interface specified entirely by a
user via a user input including one or more configuration commands provided to
the command processor at run time of the command processor and interpreted by
the command interpreter, wherein the configuration commands build graphical
objects within the graphical user interface and assign functionality to the built
graphical objects;

one or more design tools corresponding to processes within an integrated circuit design
process; and

wherein the one or more design tools operate under control of the command processor and
within the graphical user interface.

25. (Previously Presented) The integrated circuit software design suite of claim 24,
wherein the command processor interprets the user input to generate at least one graphical object
within the graphical user interface associated with at least one design tool of the one or more
design tools.

26. (Previously Presented) The integrated circuit software design suite of claim 25,
wherein the graphical object is selectable by the user to load the at least one design tool into the

graphical user interface, the graphical user interface accessible by the user to produce an integrated circuit design.

27. (Previously Presented) The command processor of claim 5, wherein at least one of the graphical objects is associated with at least one integrated design tool of the suite of integrated circuit design tools.

28. (Previously Presented) The command processor of claim 27, wherein the at least one integrated design tool is executable by the command interpreter to design and test an integrated circuit layout, and wherein the at least one of the graphical objects is selectable by the user to access the at least one integrated design tool.